**Q1.Explain IP Datagram frame format with its various fields.**

Ans-VERSION: Version of the IP protocol (4 bits), which is 4 for IPv4

HLEN: IP header length (4 bits), which is the number of 32 bit words in the header. The minimum value for this field is 5 and the maximum is 15.

Type of service: Low Delay, High Throughput, Reliability (8 bits)

Total Length: Length of header + Data (16 bits), which has a minimum value 20 bytes and the maximum is 65,535 bytes.

Identification: Unique Packet Id for identifying the group of fragments of a single IP datagram (16 bits)

Flags: 3 flags of 1 bit each : reserved bit (must be zero), do not fragment flag, more fragments flag (same order)

Fragment Offset: Represents the number of Data Bytes ahead of the particular fragment in the particular Datagram. Specified in terms of number of 8 bytes, which has the maximum value of 65,528 bytes.

Time to live: Datagram’s lifetime (8 bits), It prevents the datagram to loop through the network by restricting the number of Hops taken by a Packet before delivering to the Destination.

Protocol: Name of the protocol to which the data is to be passed (8 bits)

Header Checksum: 16 bits header checksum for checking errors in the datagram header

Source IP address: 32 bits IP address of the sender

Destination IP address: 32 bits IP address of the receiver

Option: Optional information such as source route, record route. Used by the Network administrator to check whether a path is working or not.

Due to the presence of options, the size of the datagram header can be of variable length (20 bytes to 60 bytes).

**Q2. Differentiate between the followings:**

A.Fragmentation and Reassembly-

Ans-As the name implies, IP fragmentation occurs when the receiving system cannot handle a datagram in its full form (limited MTU), and therefore the datagram is fragmented to accommodate the recipient MTU. According to Module 1, an IP packet is broken down into smaller pieces if the packet size exceeds the data link layer protocol limits. This is commonly known as fragmentation, and the process can take place at the originating device or intermediate routers. In order to retrieve the original message, the packet must be reassembled at the destination device. Intermediate routers can fragment packets, but it cannot reassemble them because fragments do not always take the same routes from source to destination.

In contrast, IP reassembly occurs at the final recipient of the message, after all fragmented datagrams have taken whatever lowest cost path was available to them have arrived. Attempting to do the latter at an intermediate step has a few challenges. For one, fragmented datagrams do not have to follow the same path to the destination – they will typically take the lowest cost path as dictated by their routing protocol, network topology, etc. In this case, no single router would necessarily have all the datagrams necessary to reassemble the entire message. Secondly, per how IP reassembly works at the destination, a router would have to subsequently “wait” for all datagrams to arrive before attempting to reassemble, which would slow down the forwarding capability of the router significantly, and any other traffic that it is processing as well. Lastly, because routers are designed to forward IP datagrams, to have them reassemble would require a more robust design to account for not only forwarding, but assembling before forwarding.

B.Multiplexing and Demultiplexing-

Ans-Multiplexing –

Gathering data from multiple application processes of sender, enveloping that data with header and sending them as a whole to the intended receiver is called as multiplexing.

Demultiplexing –

Delivering received segments at receiver side to the correct app layer processes is called as demultiplexing.

For sending data from an application at sender side to an application at the destination side, sender must know the IP address of destination and port number of the application (at the destination side) to which he want to transfer the data.

Let us consider two messaging apps that are widely used now a days viz. Hike and whatsapp. Suppose A is the sender and B is the receiver. Both sender and receiver have these applications installed in their system (say smartphone). Suppose A want to send messages to B in whatsapp and hike both. In order to do so, A must mention the IP address of B and destination port number of the whatsapp while sending the message through whatsapp application. Similarly, for the later case, A must mention the IP address of B and destination port number of the hike while sending the message.

Now the messages from both the apps will be wrapped up along with appropriate headers(viz. source IP address, destination IP address, source port no,destination port number) and sent as a single message to the receiver. This process is called multiplexing. At the destination, received message is unwrapped and constituent messages (viz messages from hike and whatsapp application) are sent to appropriate application by looking to the destination the port number. This process is called demultiplexing. Similarly, B can also transfer the messages to A.